ESD TR-68-404

COMPUTER CORPORATION OF AMERICA

ESD-TR-68-404 ESLFE

FUNDAMENTALS OF RELATIONAL STRUCTURES

3 September 1968

COMMAND SYSTEMS DIVISION
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

(Prepared under Contract No. AF 19(628)-5939 by Computer Corporation of America, 565 Technology Square, Cambridge, Mass. 02139.)

LEGAL NOTICE

OTHER NOTICES

ESD-TR-68-404

FUNDAMENTALS OF RELATIONAL STRUCTURES

3 September 1968

COMMAND SYSTEMS DIVISION
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

(Prepared under Contract No. AF 19(628)-5939 by Computer Corporation of America, 565 Technology Square, Cambridge, Mass. 02139.)

FOREWORD

This work was conducted in support of Project 2801, Task 280115 by Computer Corporation of America, Cambridge, Massachusetts under Contract AF 19(628)-5939. The contract was monitored for the U. S. Air Force by Lt. J. B. Fraser and John B. Goodenough and was principally performed during the period 15 March 1966 to 14 November 1967. The draft of this report was submitted 11 March 1968.

Publication of this report does not constitute Air Force approval of the report's findings and conclusions. It is published only for the exchange and stimulation of ideas.

Sylvia R. Mayer
SYLVIA R. MAYER
Project Officer

William F. Heisler
WILLIAM F. HEISLER, Colonel, USAF
Chief, Command Systems Division
Directorate of Planning and Technology

Abstract

A system of notation, having very elementary syntax, is developed. It is shown by example how arbitrary information may be represented in this notation. The notation is intended for computer processing, and certain deductive procedures, suitable for computers, are outlined.

Table of Contents

Chapter 1

Foundations

1.1 Introduction

While the potential capabilities of computers are recognized as being virtually unlimited, our present-day knowledge of how to realize this potential is very rudimentary. One does not know, for example, how to program computers to translate text from one language to another; to answer complicated questions without being given detailed instructions on how to obtain the answer; to draw correct inferences from large accumulations of information.

A major source of difficulty deals with communication: how do we tell the machine something, so that it may use the information intelligently?

We have learned (after a considerable expenditure of effort) how to tell a computer the sequence of steps we want it to follow in solving some specified problem. Thus we can tell the machine "take the number in cell 722 and add it to the contents of the accumulator" or even "apply procedure XYZ to the contents of cells 10, 11, and 12 and leave the result in cell 13."

This type of communication allows us to tell the machine about the particular procedures we may want the machine to execute. There is very little known, however, about how to communicate to a machine the type of information that people generally communicate to one another, namely, non-procedural information. How do we tell a computer, for example, that Lyndon Johnson is the father of Luci Nugent?

The problem is not, incidentally, that the machines can only cope with "hard", "abstract" or "mathematical" information. Our problems only grow with the level of abstraction. How do we tell a machine that all men are mortal? Or that, for any sets A, B, and C, if A is a subset of B and B is a subset of C, then A is also a subset of C? There is, of course, no good way to do it, certainly no way so that the machine, having been told the information, can use it meaningfully in subsequent tasks.

The Relational Structures development, discussed in the present report, represents an attempt to attack some of the problems of non-procedural communication with computers. A complete solution to this problem would consist at a minimum of the following interrelated parts:

1. A non-procedural language in which it is possible to express a wide variety of (or better yet, all) information.

2. Computer techniques for efficiently storing large bodies of information expressed in the language and for efficiently retrieving portions of it.

3. Computer techniques for drawing inferences from the stored information and for answering questions pertaining to the information.

The present report addresses itself to the first of these.

1.2 Background

Our work on relational structures has been motivated by the search for general-purpose fact-retrieval techniques and principles. The work is related to research in the area of question-answering systems and has also been influenced by developments in data storage techniques and mechanical theorem-proving.

Fact-Retrieval Systems

The earliest fact-retrieval systems, such as those of Green et al. [1], Lindsay [2], Simmons [3], emphasized the use of natural language as input during one or more phases of the operation. The problem of organizing the information in a general data base was not faced, either because the problem area was quite restricted and admitted to a simple representation of facts, or because the importance of data organization was not realized.

Raphael's SIR system [4] seems to be the first with any claim to a general fact-representation scheme. Its domain of discourse is fairly non-specific compared to other systems; it deals with many sorts of objects and with some specific relations between them such as spatial position, membership, ownership, and partnership. Statements are mapped into assertions that these simple relations obtain between two objects. The state of knowledge at any time is construed as a directed-graph structure in which nodes stand for objects and connecting lines represent relations. Questions are answered by locating the relevant nodes and examining the connections between them. The system rests on a basic set of primitive predicates (such as "to the right of") which

are implemented as programs which carry out certain specific searches. Our work has differed from this approach in a way that Raphael, with some hindsight, suggested: We regard facts like "to the right of is a transitive relation" to be natural candidates for inclusion in the data base. This permits the searching strategy to be conditioned upon the type of relations in a more straightforward way.

A more recent project, Relational Data File [5] also adopts the binary relation as its fundamental means of representation. Although the subject matter is quite specific (literature in cybernetic research) the representation and searching strategies appear to be based on quite general consideration about relations. There is also a brief discussion of deriving new relations in terms of a set of primitive relations. Their approach is to consider a certain calculus of relations with operations such as converse (corresponding to our inverse) and relative product (corresponding to our chain). Our scheme is quite different, however, in that, for us, relations between relations are considered reasonable candidates for storage and retrieval. There is reason to believe (both empirically and mathematically) that any fact which can be formalized can be represented using our methods, while the limits of the Relational Data File's relational calculus are not clear.

Data Structures

Although the major part of our work has been non-specific as regards techniques for representing relations in a computer memory, it has been influenced by certain developments in this area.

Feldman's Associative Processor System [6] was designed to provide a content-addressable memory containing three-symbol words. A mixture of hash-coding and list-processing techniques were used. The content of a word, say RAB, was usually taken as an assertion that relation R obtained between A and B.

The ASP study [7] was an elaborate design for a hardware-software system aimed at fact-retrieval systems. The conceptual model used was binary relations or directed, labeled graphs. Matching a template-like graph to the data base and adding or deleting links in the area matched was the basic operation. The design introduces some interesting techniques for using distributed logic hardware to manipulate data bases that can be considered as graphs.

Deduction Procedures

Only a few systems have made attempts to do anything more than direct searches of the data base in response to questions. The type of questions answerable by such systems is usually sharply delimited and well understood.

The Relational Data File [5] study makes several reasonable suggestions for increasing the deductive capability of the system. The study deals with the problems of straightforward logical deductions and also considers the concept of plausible deductions. Interaction between the user and the system is advocated as an approach.

A recent study by Raphael [8] reports an attempt to employ the mechanical theorem-proving procedures of Robinson and others to this problem.

1.3 A Model of the Universe of Discourse

Before we can have a language, that is, a way of representing certain facts about the universe, we must have a fairly clear notion regarding the universe we wish to speak about.

The universe with which we deal in relational structures is a very simple one; it is made up of a finite number of objects and of various combinations, ordered and unordered, of these objects.

Specifically, the following statements recursively define the elements in this universe of discourse:

1. The objects $O_1, O_2, \ldots, O_n$ are elements.

2. If $e_1, \ldots, e_m$ are elements, then the ordered m-tuple $\langle e_1, \ldots, e_m \rangle$ is also an element.

3. If $e_1, \ldots, e_p$ are elements, then the set $\{e_1, \ldots, e_p\}$ is also an element. The empty set $\emptyset$ is an element.

These elements, then, are the things we may talk about. It should be noted right away that if the number n of objects is at all large, we will not have occasion to talk about very many of the possible elements. For example, if $n = 4$, the number of ordered quadruples is 256 and the number of sets of ordered quadruples is 2^{256} (about 10^{77}), and we've barely started. In fact, if we allow arbitrarily large m-tuples (with the same element appearing in more than one position) then the number of elements is infinite, even though the number of objects is

finite. This fact is of no particular interest, however, since the difference between infinite and inconceivably large is, for all practical purposes, nonexistent.

1.4 Relations

Certain elements, called relations, deserve particular attention.

We define an n-ary relation to be a set of ordered n-tuples of elements.

For example, the set of ordered pairs $<O_i, O_j>$ such that O_i is the father of O_j, defines the binary relation "x is the father of y". Or, again, the set of ordered triples $<O_i, O_j, O_k>$, such that O_j is something that O_i threw at O_k, defines the ternary relation "x threw y at z".

This definition of relation is in agreement with the definition given in set theory (a relation from A to B is a subset of $A \times B$), but it is not necessarily intuitive. The point of view taken here is that if one had a list of all father-child pairs, one would know as much as can be known regarding the definition of "x is the father of y"; one may similarly say that if one had a list of all red things or of all grimthorpes, one would know as much as could be known regarding the definition of "x is red" or "x is a grimthorpe".

In practice, of course, our lists will be incomplete. We presumably cannot list all pairs in the relation "x is the father of y", or all triples in the relations "x threw y at z", or all elements in the set of red things or of grimthorpes. One would say, then, that in practice, our understanding of these concepts is necessarily incomplete.

1.5 Relational Expressions and Relational Structures

A symbol is any string of characters, the string being formed from an alphabet which includes space but excludes parentheses and comma. Examples:

> A
>
> 3x = 17y
>
> Which way did he go?

A relational expression (RE) is a string consisting of a left parthenthesis, a symbol or else several symbols separated by commas, and a right parenthesis. Examples:

> (A, B, C)
>
> (King of Siam)
>
> (alpha, beta, George)

A relational structure (RS) is a set of relational expressions.

1.6 The Interpretation of Relational Expressions

1.6.1 Interpretation of Relational Expressions as Sentences

We define a mapping (or interpretation) I1 which maps REs into a certain restricted class of sentences.

We do this as follows. We start with the relational expression

$$(n_0, n_1, \ldots, n_m), \qquad m \geq 1$$

The symbols $n_0, \ldots, n_m$ are interpreted as naming entities in the real world. The symbol n_0, in particular, is interpreted as naming a certain set. The ordered m-tuple of entities named by $n_1, \ldots, n_m$ is then asserted as being a member of the set named by n_0.

Thus we may write

$$I1[(n_0, n_1, \ldots, n_m)] \Longrightarrow \langle n_1, \ldots, n_m \rangle \in n_0$$

If $m = 2$, the set n_0 is interpreted as a set of ordered pairs; i.e., n_0 is interpreted as a binary relation. If $m = 3$, n_0 is interpreted as a set of ordered 3-tuples; i.e., n_0 is interpreted as a ternary relation. Etc.

Example: Let "Father of" designate the relation "is the father of" (strictly speaking, let "Father of" be the name of the set having as members ordered pairs of objects, the first of each pair being the father of the second). Let "Lyndon Johnson" designate Lyndon Johnson, and let "Luci Nugent" designate Luci Nugent. Then the RE

(Father of, Lyndon Johnson, Luci Nugent) (1)

is interpreted by I1 as saying that the ordered pair consisting, first, of Lyndon Johnson and, second, of Luci Nugent, is a member of the binary relation called "Father of". Thus (1) is translated by I1 into the sentence

$\langle$Lyndon Johnson, Luci Nugent$\rangle \in$ Father of (2)

Now, (2) is a sentence in the sense of asserting something true or false about the real world. However, the notation of (2) is that of formal mathematics. Sentence (2) can be expressed in English as:

> The ordered pair consisting, first, of Lyndon Johnson and, second, of Luci Nugent, is a member of the set of ordered pairs of objects, the first of each pair being the father of the second. (3)

We may note that (3) corresponds in a fairly obvious way to the colloquial English sentence:

Lyndon Johnson is the father of Luci Nugent. (4)

1.6.2 Interpretation of Sentences as Relational Structures

Let Σ be the set of sentences of the form

$$\langle n_1, \ldots, n_m \rangle \in n_0$$

where $m \geq 1$ and $n_0,\ldots,n_m$ are symbols. We have seen above how relational expressions may be translated into members of Σ by I1.

Let Σ' be the set of English sentences corresponding in meaning to the sentences of Σ.

We now assert the following conjecture, which is fundamental to much of the work on relational structures.

Conjecture: Any meaningful English sentence can be expressed as a set of sentences in Σ'.

The process of translating an English sentence into a relational structure can then (in principle) be achieved as follows.

1. Express the given sentence as one or more sentences in Σ'.

2. Express each sentence in Σ' obtained in step 1 as a sentence in Σ.

3. Express each sentence in Σ obtained in step 2 above as that relational expression which maps into the sentence by I1. The set of REs so obtained is the relational structure corresponding to the original sentence.

1.6.3 Sentences about Relational Expressions and Sentences

As an exercise, we consider how, under I1, relational structures may express facts about relational expressions and about sentences.

Suppose, e.g., that we wish to say that a particular RE is a RE. Let "SRE" be the name of the set of relational expressions and let "RE1" be the name of a particular RE. We can then write the RE

(SRE, RE1) (5)

By I1, (5) says that RE1 is a relational expression.

Suppose now we want to express:

> The sentence "Luci Nugent is the father of Lyndon Johnson" is false.

Let "SENTENCEX" be the name of the sentence "Luci Nugent is the father of Lyndon Johnson". Let "FALSE" be the name of the set of false sentences. Then write the RE

(FALSE, SENTENCEX)

Suppose now we wish to express the fact that "SENTENCEX" is the name of the sentence "Luci Nugent is the father of Lyndon Johnson". Let "NAME" be the name of the relation that exists between a name and the thing named, and let "QUOTESENTENCEX" be the name of the symbol "SENTENCEX". The desired RE is:

(NAME, QUOTESENTENCEX, SENTENCEX)

Finally, suppose we wish to express the fact that "Luci Nugent is the father of Lyndon Johnson" corresponds to the RE (Father, Luci Nugent, Lyndon Johnson) by I1.

Let "REX" be the name of the RE (Father of, Luci Nugent, Lyndon Johnson), and let "INTERP1" be the name of the relation that exists between an English sentence and the RE to which the English sentence corresponds by I1. Then we can write

(INTERP1, SENTENCEX, REX)

1.6.4 Expressing Functions as Expressions Interpretable by I1

An interesting question is whether we can express functions as relational structures in such a way that the English sentence images of the relational structures under I1 correctly express the given function. It would seem to us that this can always be done.

However, it is not clear that it is important to express functions in this way. It may be much simpler to have another interpretation I2 which maps relational structures directly into executable algorithms. Naturally, if one tries to interpret by I2 the expressions that were meant to be interpreted by I1, or vice-versa, one will get nonsense. However, this need be no more frightening than the possibility that a program will start executing data.

1.7 Five Short Examples

The Universe of Discourse

For the purpose of our examples, we assume we have some indeterminate number of objects, $O_1, \ldots, O_n$.

Two of these have specific names, namely LYNDON JOHNSON and LUCI NUGENT.

From these objects, we have formed a number of sets of objects, three of which have specific names:

FATHERS
COF
MEN

The set FATHERS contains objects which are fathers. The set COF contains objects which are children of fathers. And the set MEN contains men. We assume further that sets of ordered pairs of objects have been formed. One of these is called FATHER OF, and consists of all pairs in which the first is father of the second.

We assume further that sets of ordered pairs of arbitrary elements have been formed (see Section 1.3 for the definition of <u>element</u>). Three of these are called SUBSET, LH and RH. The set called SUBSET consists of ordered pairs in which the first member is a set which is contained in the second member.

The set called LH (an abbreviation for "left half") consists of ordered pairs in which the first member is a set S and the second

is a binary relation R, and where S is identical to the set formed by taking the first members of the pairs in R.

For example, let the relation R be DAUGHTER OF; that is, let R be the set of ordered pairs <x,y> where x is the daughter of y. Let S be the set of elements appearing as first members in the pairs <x,y>. S is then the set of daughters. The ordered pair <S,R> is then a member of LH.

The set called RH (an abbreviation for "right half") consists of ordered pairs in which the first member is a set S and the second is a relation R; and where S is identical to the set formed by taking the second members of the pairs in R.

For example, let the relation R (as above) be DAUGHTER OF; that is, let R be the set of ordered pairs <x,y> where x is the daughter of y. Let T be the set of elements appearing as second members in the pairs <x,y>. T is then the set of parents of daughters. The ordered pair <T,R> is then a member of RH.

Example 1

(FATHER OF, LYNDON JOHNSON, LUCI NUGENT)

By I1, this is interpreted as saying that the ordered pair of elements denoted by <LYNDON JOHNSON, LUCI NUGENT> is a member of the set (or relation) denoted by FATHER OF. In short, it says that Lyndon Johnson is the father of Luci Nugent.

Example 2

(SUBSET, FATHERS, MEN)

By I1, this is interpreted as saying that the ordered pair of elements denoted by

<FATHERS, MEN>

is a member of the set (or relation) denoted by SUBSET; in short, that the set of fathers is contained in the set of men; in other words, that all fathers are men.

Example 3

(SUBSET, MEN, COF)

By the same argument, this means that all men are children of fathers, or in other words, that every man has a father.

Example 4

(LH, FATHERS, FATHER OF)

This says that the pair

<FATHERS, FATHER OF>

stand in the relation LH. That is to say, fathers are those things which are first members of the pairs in the relation FATHER OF.

Example 5

(RH, COF, FATHER OF)

This says that the pair

<COF, FATHER OF>

stands in the relation RH. That is to say, members of COF (i.e., children of fathers) are those things which are the second members of the pairs in the relation FATHER OF.

Chapter 2
An Extended Example

2.1 Introduction

In the present chapter we deal with the relational expressions representing the first eight sentences of A Student Guide to Russia*. Section 2.2 discusses some fundamental relations. Section 2.3 presents the full text. Section 2.4 presents the same text, but broken down into simple sentences which are numbered. Section 2.5 presents each simple sentence of Section 2.4 along with the corresponding set of relational expressions. Section 2.6 presents an alphabetical listing of all relational expressions, with references to the sentences in which they occur. Finally, Section 2.7 presents an alphabetical index of symbols, cross-referenced to the expressions and sentences in which they occur.

It should be stated that this represents a first attempt at such a task. A second go-around would probably produce much improved results.

---

* Robert Katz, A Student Guide to Russia and Eastern Europe, Monarch Press, New York, 1965.

2.2 Some Fundamental Relations

The relations LH (left half) and RH (right half) were discussed informally in Section 1.7. The formal definition is as follows.

The left half (LH) of a relation R consists of the set of elements which are left members of pairs in R. More precisely, we say that

A is the LH of R
(written (LH,A,R))

means that

$$(x)[X \epsilon A \equiv (\exists y)(Rxy)]$$

The relation right half (RH) is analogously defined.

The left intersection (LI) of a relation R and a set S is the set of those pairs in R whose left members are members of S. Thus, the relation BROTHER OF is the LI of the relation SIBLING OF and the set MALE.

More precisely, we say that

A is the LI of R and S
(written (LI,A,R,S))

means that

$$(x)(y)[\langle Axy\rangle \equiv \langle Rxy\rangle \wedge x\epsilon S]$$

The relation right intersection (RI) is analogously defined.

The relation miniset obtains between two sets, A and B, and an element O, if A is like B except for having O as a member. Thus we might say that RUS is the miniset of Ø (the empty set) and Russia, to mean that RUS is a set containing Russia as its only member.

More precisely, we say that

A is the MINISET of B and C
(written (MINISET,A,B,C))

means that

$$(x)(x\epsilon A \equiv x\epsilon B \vee x = C)$$

2.3 The English Text

The Union of Soviet Socialist Republics is the biggest country in the world. Stretching across eleven time zones, it is 2.5 times the size of the United States, 7 times the size of India, 60 times the size of Japan, and almost as wide as the diameter of the earth. The sun never sets on the Soviet Union. When darkness falls on its western borders, dawn is breaking over Vladivostok. Lying only a dozen miles or so from the Alaskan frontier of the United States, the USSR is touched by 12 countries of Asia and Europe, and twelve seas of the Atlantic, the Arctic and Pacific oceans. It is a country of harsh tundra and fiery deserts, of angry volcanoes and impassible forests, of rugged mountains and endless plains, of 100,000 rivers and every type of known mineral in the universe.

It is a country of 225,000,000 people, of more than 100 different nationalities: Russians and Ukranians, Moldavians and Karelians, Komi and Nentsi, Tatars, Udmurts, Bashkirs, Azerbaijanians, Ingushes, Kabardinians, Kazakhs, Uzbeks--people among whom some have traveled in outer space and some still drive an ox-cart.

There are many reasons to visit the USSR, and certainly the best of these is not because it's big and diverse, nor because it is a land of many contrasts.

2.4 Text Broken Up into Simple Sentences

The Union of Soviet Socialist Republics is the biggest country in the world$_{1.1}$. The USSR stretches across eleven time zones$_{2.1}$. The USSR is 2.5 times the size of the USA$_{2.2}$. The USSR is 7 times the size of India$_{2.3}$. The USSR is 60 times the size of Japan$_{2.4}$. The USSR is almost as wide as the diameter of the earth$_{2.5}$. At any given time, some object in the USSR is sunlit$_{3.1}$. When darkness falls on the western borders of the USSR, dawn is breaking over Vladivostok$_{4.1}$. The USSR lies only a dozen miles or so from the Alaskan frontier of the United States$_{5.1}$. The USSR is touched by 12 countries of Asia and Europe, 12 seas of the Atlantic, the Arctic and the Pacific oceans$_{5.2}$.

The USSR is a country of harsh tundra$_{6.1}$. The USSR is a country of fiery deserts$_{6.2}$. The USSR is a country of angry volcanoes$_{6.3}$. The USSR is a country of impassible forests$_{6.4}$. The USSR is a country of rugged mountains$_{6.5}$. The USSR is a country of endless plains$_{6.6}$. The USSR is a country of 100,000 rivers$_{6.7}$. The USSR is a country of every type of known mineral in the universe$_{6.8}$.

It is a country of 225,000,000 people, of more than 100 different nationalities: Russians and Ukranians, Moldavians and Karelians, Komi and Nentsi, Tatars, Udmurts, Bashkirs, Azerbaijanians, Ingushes, Kabardinians, Kazakhs, Uzbeks--people among whom some have traveled in outer space and some still drive an ox-cart$_{7.1}$.

There are many reasons for the reader visits the USSR$_{8.1}$. The fact that the USSR is big and diverse is not one of the best reasons for the readers visiting the USSR$_{8.2}$. The fact that the USSR is a land of many contrasts is not one of the best reasons for the readers visiting the USSR$_{8.3}$.

2.5 The Corresponding Relational Expressions

Sentence 1.1:

"The Union of Soviet Socialist Republics is the biggest country in the world."

Structure:
(1) (LI, Q2, BIGGER, USSR)
(2) (RH, A1, Q2)
(3) (SUBSET, COUNTRIES, A1)

Sentence 2.1:

"The USSR stretches across eleven time zones."

Structure:
(4) (WIDTH OF, WIDTH OF USSR, USSR)
(5) (RI, Y2, WIDTH OF, TIME ZONES)
(6) (LH, Z2, Y2)
(7) (MINISET, Z2, NULL SET, W2)
(8) (QUOTIENT OF, ELEVEN, WIDTH OF USSR, W2)

Sentence 2.2:

"The USSR is 2.5 times the size of the USA."

Structure:
(9) (QUOTIENT OF, 2.5, B2, C1)
(10) (SIZE OF, C1, USA)
(11) (SIZE OF, B2, USSR)

Sentence 2.3:

"The USSR is 7 times the size of India."

Structure: (12) (QUOTIENT OF, 7, B2, E1)
(11)* (SIZE OF, B2, USSR)
(13) (SIZE OF, E1, INDIA)

Sentence 2.4:

"The USSR is 60 times the size of Japan."

Structure: (14) (QUOTIENT OF, 60, B2, G1)
(11)* (SIZE OF, B2, USSR)
(15) (SIZE OF, G1, JAPAN)

Sentence 2.5:

"The USSR is almost as wide as the diameter of the earth."

Structure: (16) (WDITH OF, H, USSR)
(17) (DIAMETER OF, I, EARTH)
(18) (ALMOST EQUAL, H, I)

Sentence 3.1:

"At any given time, some object in the USSR is sunlit."

---

* Expression has occurred previously.

Structure: (19) (GEOIN, TIR, USSR)

(20) (LI, X2, SUNLIT, TIR)

(21) (RH, Y3, X2)

(22) (SUBSET, TIMES, Y3)

Sentence 4.1:

"When darkness falls on its western borders, dawn is breaking over Vladivostok."

Structure: (23) (SUPERLATIVE, WESTERN-MOST, WEST OF)

(24) (WESTERN-MOST, WBR, TIR)

(25) (GEOIN, TIV, VLADIVOSTOK)

(26) (LI, X3, SUNRISE, TIV)

(27) (RH, Y4, X3)

(28) (LI, W3, SUNSET, WBR)

(29) (RH, Z3, W3)

(30) (SUBSET, Z3, Y4)

Sentence 5.1:

"The USSR lies only a dozen miles or so from the Alaskan frontier of the United States."

Structure: (31) (FRONTIER, Z4, USA)

(32) (GEOIN, Y5, Z4)

(33) (GEOIN, X4, ALASKA)

(34) (INTERSECT, W4, (X4, Y5))

(35) (LAMBDA2, A2, DISTMILES, W4)

(36) (LAMBDA3, B3, A2, LMR)

(37) (RHO1, C2, B3)

(38) (LI, D1, LESS THAN OR EQUAL, (12))
(39) (RH, E2, D1)
(40) (RI, F1, LESS THAN OR EQUAL, (13))
(41) (LH, G2, F1)
(42) (INTERSECT, H1, (G2, E2))
(43) (SUBSET, C2, H1)

Sentence 5.2:

"The USSR is touched by 12 countries of Asia and Europe and 12 seas of the Atlantic, the Arctic and the Pacific oceans."

Structure:
(44) (GEOIN, TIA, ASIA)
(45) (GEOIN, TIE, EUROPE)
(46) (GEOIN, ATL, ATLANTIC OCEAN)
(47) (GEOIN, ART, ARCTIC OCEAN)
(48) (GEOIN, PAC, PACIFIC OCEAN)
(49) (UNION, A3, TIA, TIE)
(50) (UNION, B4, ATL, ART, PAC)
(51) (SUBSET, C3, COUNTRIES)
(52) (SUBSET, C3, A3)
(53) (SUBSET, D2, SEAS)
(54) (SUBSET, D2, B4)
(55) (CARDINALITY, 12, C3)
(56) (CARDINALITY, 12, D2)
(57) (LI, E3, TOUCHES, USSR)
(58) (RH, F2, E3)
(59) (SUBSET, C3, F2)
(60) (SUBSET, D2, F2)

Sentence 6.1:

"The USSR is a country of harsh tundra."

Structure: (61) (INTERSECT, X1, (TUNDRA, HARSH, TIR))
(62) (NOT EQUAL, X1, NULL SET)

Sentence 6.2:

"The USSR is a country of fiery deserts."

Structure: (63) (INTERSECT, Y1, (DESERTS, FIERY, TIR))
(64) (NOT EQUAL, Y1, NULL SET)

Sentence 6.3:

"The USSR is a country of angry volcanoes."

Structure: (65) (INTERSECT, Z1, (VOLCANOES, ANGRY, TIR))
(66) (NOT EQUAL, Z1, NULL SET)

Sentence 6.4:

"The USSR is a country of impassible forests."

Structure: (67) (INTERSECT, W1, (FORESTS, IMPASSIBLE, TIR))
(68) (NOT EQUAL, W1, NULL SET)

Sentence 6.5:

"The USSR is a country of rugged mountains."

Structure: (69) (INTERSECT, V1, (MOUNTAINS, RUGGED, TIR))
(70) (NOT EQUAL, V1, NULL SET)

Sentence 6.6:

"The USSR is a country of endless plains."

Structure: (71) (INTERSECT, T, (PLAINS, ENDLESS, TIR))
(72) (NOT EQUAL, T, NULL SET)

Sentence 6.7:

"The USSR is a country of 100,000 rivers."

Structure: (73) (INTERSECT, S, (RIVERS, TIR))
(74) (CARDINALITY, 100000, S)

Sentence 6.8:

"The USSR is a country of every type of known mineral in the universe."

Structure: (75) (LH, N1, (MINERAL TYPE, TIR))
(76) (INTERSECT, Q1, (N1, KNOWN))
(77) (RI, R, MINERAL TYPE, TIR)
(78) (LH, P1, R)
(79) (SUBSET, Q1, P1)

Sentence 7.1:

"It is a country of 225,000,000 people, of more than 100 different nationalities: Russians and Ukranians, Moldavians and Karelians, Komi and Nentsi, Tatars, Udmurts, Bashkirs, Azerbaijanians, Ingushes, Kabardinians, Kazakhs, Uzbeks--people among whom some have traveled in outer space and some still drive an ox-cart."

Structure:

(80) (INTERSECT, P2, (LMR, SET OF ALL PEOPLE))
(81) (CARDINALITY, 225000000, P2)
(82) (RI, A4, NATIONALITY, P2)
(83) (LH, N2, A4)
(84) (CARDINALITY, M, N2)
(85) (LESS THAN OR EQUAL, 100, M)
(86) (N2, RUSSIANS)
(87) (N2, UKRANIANS)
(88) (N2, MOLDAVIANS)
(89) (N2, KARELIANS)
(90) (N2, KOMI)
(91) (N2, NENTSI)
(92) (N2, TATARS)
(93) (N2, UDMURTS)
(94) (N2, BASHKIRS)
(95) (N2, AZERBAIJANIANS)
(96) (N2, INGUSHES)
(97) (N2, KABARDINIANS)
(98) (N2, KAZAKHS)
(99) (N2, UZBEKS)
(100) (INTERSECT, B5, (P2, OUTERSPACE TRAVELERS))
(101) (NOT EQUAL, B5, NULL SET)
(102) (INTERSECT, D3, (P2, OX-CART DRIVERS))
(103) (NOT EQUAL, D3, NULL SET)

Sentence 8.1:

"There are many reasons for the reader visits the USSR."

Structure: (104) (MEANING, (X5), Y6)
(105) (FIRST, VISITS, X5)
(106) (SECOND, READER, X5)
(107) (THIRD, USSR, X5)
(108) (RI, Z5, REASON FOR, (Y6))
(109) (LH, W5, Z5)
(110) (MANY, W5)

Sentence 8.2:

"The fact that the USSR is big and diverse is not one of the best reasons for the readers visiting the USSR. The fact that the USSR is a land of many contrasts is not one of the best reasons for the readers visiting the USSR."

Structure: (111) (FIRST, BIG, A5)
(112) (SECOND, USSR, A5)
(113) (FIRST, DIVERSE, B6)
(114) (SECOND, USSR, B6)
(115) (MINISET, C4, MULL SET, A5)
(116) (MINISET, D4, C4, B6)
(117) (FIRST, INTERSECT, E4)
(118) (SECOND, Q3, E4)
(119) (THIRD, (CONTRASTS, TIR), E4)
(120) (FIRST, MANY, F3)
(121) (SECOND, Q3, F3)
(122) (MINISET, G3, NULL SET, E4)

(123) (MINISET, H2, G3, F3)

(124) (COMPARATIVE, BRF, REASON FOR)

(125) (SUPERLATIVE, BSTRF, BRF)

(126) (RI, J, BSTRF, Z5)

(127) (LH, K, J)

(128) (LH, N3, K)

(129) (COMPLEMENT, NN3, N3)

(130) (NN3, D4)

(131) (NN3, H3)

2.6 The Relational Expressions Alphabetized and Cross-Referenced to Sentences

Expression No.	Expression	Simple Sentence No(s).
(18)	(ALMOST EQUAL, H, I)	2.5
(84)	(CARDINALITY, M, N2)	7.1
(74)	(CARDINALITY, 100000, S)	6.7
(55)	(CARDINALITY, 12, C3)	5.2
(56)	(CARDINALITY, 12, D2)	5.2
(81)	(CARDINALITY, 225000000, P2)	7.1
(124)	(COMPARATIVE, BRF, REASON FOR)	8.2
(129)	(COMPLEMENT, NN3, N3)	8.2
(17)	(DIAMETER OF, I, EARTH)	2.5
(111)	(FIRST, BIG, A5)	8.2
(113)	(FIRST, DIVERSE, B6)	8.2
(117)	(FIRST, INTERSECT, E4)	8.2
(120)	(FRIST, MANY, F3)	8.2
(105)	(FIRST, VISITS, X5)	8.1
(31)	(FRONTIER, Z4, USA)	5.1
(47)	(GEOIN, ART, ARCTIC OCEAN)	5.2
(46)	(GEOIN, ATL, ALTANTIC OCEAN)	5.2
(48)	(GEOIN, PAC, PACIFIC OCEAN)	5.2
(44)	(GEOIN, TIA, ASIA)	5.2
(45)	(GEOIN, TIE, EUROPE)	5.2
(19)	(GEOIN, TIR, USSR)	3.1
(25)	(GEOIN, TIV, VLADIVOSTOK)	4.1
(33)	(GEOIN, X4, ALASKA)	5.1
(32)	(GEOIN, Y5, Z4)	5.1
(100)	(INTERSECT, B5, (P2, OUTER SPACE TRAVELERS))	7.1

Expression No.	Expression	Simple Sentence No(s).
(102)	(INTERSECT, D3, (P2, OX-CART DRIVERS))	7.1
(42)	(INTERSECT, H1, (G2, E2))	5.1
(80)	(INTERSECT, P2, (LMR, SET OF ALL PEOPLE))	7.1
(76)	(INTERSECT, Q1, (N1, KNOWN))	6.8
(73)	(INTERSECT, S, (RIVERS, TIR))	6.7
(71)	(INTERSECT, T, (PLAINS, ENDLESS, TIR))	6.6
(69)	(INTERSECT, V1, (MOUNTAINS, RUGGED, TIR))	6.5
(67)	(INTERSECT, W1, (FORESTS, IMPASSIBLE, TIR))	6.4
(34)	(INTERSECT, W4, (X4, Y5))	5.1
(61)	(INTERSECT, X1, (TUNDRA, HARSH, TIR))	6.1
(63)	(INTERSECT, Y1, (DESERTS, FIERY, TIR))	6.2
(65)	(INTERSECT, Z1, (VOLCANOES, ANGRY, TIR))	6.3
(35)	(LAMBDA2, A2, DISTMILES, W4)	5.1
(36)	(LAMBDA3, B3, A2, LMR)	5.1
(85)	(LESS THAN OR EQUAL, 100, M)	7.1
(41)	(LH, G2, F1)	5.1
(127)	(LH, K, J)	8.2
(75)	(LH, N1, (MINERAL TYPE, TIR))	6.8
(83)	(LH, N2, A4)	7.1

Expression No.	Expression	Simple Sentence No(s).
(128)	(LH, N3, K)	8.2
(78)	(LH, P1, R)	6.8
(109)	(LH, W5, Z5)	8.1
(6)	(LH, Z2, Y2)	2.1
(38)	(LI, D1, LESS THAN OR EQUAL, (12))	5.1
(57)	(LI, E3, TOUCHES, USSR)	5.2
(1)	(LI, Q2, BIGGER, USSR)	1.1
(28)	(LI, W3, SUNSET, WBR)	4.1
(20)	(LI, X2, SUNLIT, TIR)	3.1
(26)	(LI, X3, SUNRISE, TIV)	4.1
(110)	(MANY, W5)	8.1
(104)	(MEANING, (X5), Y6)	8.1
(115)	(MINISET, C4, NULL SET, A5)	8.2
(116)	(MINISET, D4, C4, B6)	8.2
(122)	(MINISET, G3, NULL SET, E4)	8.2
(123)	(MINISET, H3, G3, F3)	8.2
(7)	(MINISET, Z2, NULL SET, W2)	2.1
(130)	(NN3, D4)	8.2
(131)	(NN3, H3)	8.2
(101)	(NOT EQUAL, B5, NULL SET)	7.1
(103)	(NOT EQUAL, D3, NULL SET)	7.1
(72)	(NOT EQUAL, T, NULL SET)	6.6
(70)	(NOT EQUAL, V1, NULL SET)	6.5
(68)	(NOT EQUAL, W1, NULL SET)	6.4
(62)	(NOT EQUAL, X1, NULL SET)	6.1
(64)	(NOT EQUAL, Y1, NULL SET)	6.2
(66)	(NOT EQUAL, Z1, NULL SET)	6.3
(95)	(N2, AZERBAIJANIANS)	7.1

Expression No.	Expression	Simple Sentence No(s).
(94)	(N2, BASHKIRS)	7.1
(96)	(N2, INGUSHES)	7.1
(97)	(N2, KABARDINIANS)	7.1
(89)	(N2, KARELIANS)	7.1
(98)	(N2, KAZAKHS)	7.1
(90)	(N2, KOMI)	7.1
(88)	(N2, MOLDAVIANS)	7.1
(91)	(N2, NENTSI)	7.1
(86)	(N2, RUSSIANS)	7.1
(92)	(N2, TATARS)	7.1
(93)	(N2, UDMURTS)	7.1
(87)	(N2, UKRANIANS)	7.1
(99)	(N2, UZBEKS)	7.1
(8)	(QUOTIENT OF, ELEVEN, WIDTH OF USSR, W2)	2.1
(9)	(QUOTIENT OF, 2.5, B2, C1)	2.2
(12)	(QUOTIENT OF, 7, B2, E1)	2.3
(14)	(QUOTIENT OF, 60, B2, G1)	2.4
(2)	(RH, A1, Q2)	1.1
(39)	(RH, E2, D1)	5.1
(58)	(RH, F2, E3)	5.2
(21)	(RH, Y3, X2)	3.1
(27)	(RH, Y4, X3)	4.1
(29)	(RH, Z3, W3)	4.1
(37)	(RHO1, C2, B3)	5.1
(82)	(RI, A4, NATIONALITY, P2)	7.1
(40)	(RI, F1, LESS THAN OR EQUAL, (13))	5.1

Expression No.	Expression	Simple Sentence No(s).
(126)	(RI, J, BSTRF, Z5)	8.2
(77)	(RI, R, MINERAL TYPE, TIR)	6.8
(5)	(RI, Y2, WIDTH OF, TIME ZONES)	2.1
(108)	(RI, Z5, REASON FOR, (Y6))	8.1
(118)	(SECOND, Q3, E4)	8.2
(121)	(SECOND, Q3, F3)	8.2
(106)	(SECOND, READER, X5)	8.1
(112)	(SECOND, USSR, A5)	8.2
(114)	(SECOND, USSR, B6)	8.2
(11)	(SIZE OF, B2, USSR)	2.2, 2.3, 2.4
(10)	(SIZE OF, C1, USA)	2.2
(13)	(SIZE OF, E1, INDIA)	2.3
(15)	(SIZE OF, G1, JAPAN)	2.4
(3)	(SUBSET, COUNTRIES, A1)	1.1
(43)	(SUBSET, C2, H1)	5.1
(52)	(SUBSET, C3, A3)	5.2
(51)	(SUBSET, C3, COUNTRIES)	5.2
(59)	(SUBSET, C3, F2)	5.2
(54)	(SUBSET, D2, B4)	5.2
(60)	(SUBSET, D2, F2)	5.2
(53)	(SUBSET, D2, SEAS)	5.2
(22)	(SUBSET, TIMES, Y3)	3.1
(79)	(SUBSET, Q1, P1)	6.8
(30)	(SUBSET, Z3, Y4)	4.1
(125)	(SUPERLATIVE, BSTRF, BRF)	8.2
(23)	(SUPERLATIVE, WESTERN-MOST, WEST OF)	4.1

Expression No.	Expression	Simple Sentence No(s).
(119)	(THIRD, (CONTRASTS, TIR), E4)	8.2
(107)	(THIRD, USSR, X5)	8.1
(49)	(UNION, A3, TIA, TIE)	5.2
(50)	(UNION, B4, ATL, ART, PAC)	5.2
(24)	(WESTERN-MOST, WBR, TIR)	4.1
(16)	(WIDTH OF, H, USSR)	2.5
(4)	(WIDTH OF, WIDTH OF USSR, USSR)	2.1

2.7 Alphabetical List of Symbols, Cross-Referenced to Sentences and Expressions

Symbol	Expression No(s).	Simple Sentence No(s).
ALASKA	33	5.1
ALMOST EQUAL	18	2.5
ANGRY	65	6.3
ARCTIC OCEAN	47	5.2
ART	47, 50	5.2
ASIA	44	5.2
ATL	46, 50	5.2
ATLANTIC OCEAN	46	5.2
AZERBAIJANIANS	95	7.1
A1	2, 3	1.1
A2	35, 36	5.1
A3	49, 52	5.2
A4	82, 83	7.1
A5	111, 112, 115	8.2
BASHKIRS	94	7.1
BIG	111	8.2
BIGGER	1	1.1
BRF	124, 125	8.2
BSTRF	125, 126	8.2
B2	9, 11, 12, 14,	2.2, 2.3, 2.4
B3	36, 37	5.1
B4	50, 54	5.2
B5	100, 101	7.1
B6	113, 114, 116	8.2

Symbol	Expression No(s).	Simple Sentence No(s).
CARDINALITY	55, 56, 74, 81 84	5.2, 6.7, 7.1
COMPARATIVE	124	8.2
COMPLEMENT	129	8.2
CONTRASTS	119	8.2
COUNTRIES	3, 51	1.1, 5.2
C1	9, 10	2.2
C2	37, 43	5.1
C3	51, 52, 55, 59	5.2
C4	115, 116	8.2
DESERTS	63	6.2
DIAMETER OF	17	2.5
DISTMILES	35	5.1
DIVERSE	113	8.2
D1	38, 39	5.1
D2	53, 54, 56, 60	5.2
D3	102, 103	7.1
D4	116, 130	8.2
EARTH	17	2.5
ELEVEN	8	2.1
ENDLESS	71	6.6
EUROPE	45	5.2
E1	12, 13	2.3
E2	39, 42	5.1
E3	57, 58	5.2
E4	117, 118, 119, 122	8.2

Symbol	Expression No(s).	Simple Sentence No(s).
FIERY	63	6.2
FIRST	105, 111, 113, 117, 120	8.1, 8.2
FORESTS	67	6.4
FRONTIER	31	5.1
F1	40, 41	5.1
F2	58, 59, 60	5.2
F3	120, 121, 123	8.2
GEOIN	19, 25, 32, 33, 44, 45, 46, 47, 48	3.1, 4.1, 5.1, 5.2
G1	14, 15	2.4
G2	41, 42	5.1
G3	122, 123	8.2
H	16, 18	2.5
HARSH	61	6.1
H1	42, 43	5.1
H3	123, 131	8.2
I	17, 18	2.5
IMPASSIBLE	67	6.4
INDIA	13	2.3
INGUSHES	96	7.1
INTERSECT	34, 42, 61, 63, 65, 67, 69, 71, 73, 76, 80, 100, 102, 117	5.1, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 7.1, 8.2

Symbol	Expression No(s).	Simple Sentence No(s).
J	126, 127	8.2
JAPAN	15	2.4
K	127, 128	8.2
KABARDINIANS	97	7.1
KARELIANS	89	7.1
KAZAKHS	98	7.1
KNOWN	76	6.8
KOMI	90	7.1
LAMBDA2	35	5.1
LAMBDA3	36	5.1
LESS THAN OR EQUAL	38, 40, 85	5.1, 7.1
LH	6, 41, 75, 78, 83, 109, 127, 128	2.1, 5.1, 6.8, 7.1, 8.1, 8.2
LI	1, 20, 26, 28, 38, 57	1.1, 3.1, 4.1, 5.1, 5.2
LMR	36, 80	5.1, 7.1
M	84, 85	7.1
MANY	110, 120	8.1, 8.2
MEANING	104	8.1
MINERAL TYPE	75, 77	6.8
MINISET	7, 115, 116, 122, 123	2.1 , 8.2
MOLDAVIANS	88	7.1
MOUNTAINS	69	6.5

Symbol	Expression No(s).	Simple Sentence No(s).
NATIONALITY	82	7.1
NENTSI	91	7.1
NN3 (NOT N3)	129, 130, 131	8.2
NOT EQUAL	62, 64, 66, 68, 70, 72, 101, 103	6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 7.1
NULL SET	7, 62, 64, 66, 68, 70, 72, 101, 103, 115, 122	2.1, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 7.1, 8.2
N1	75, 76	6.8
N2	83, 84, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99	7.1
N3	128, 129	8.2
OUTERSPACE TRAVELERS	100	7.1
OX-CART DRIVERS	102	7.1
PAC	48, 50	5.2
PACIFIC OCEAN	48	5.2
PLAINS	71	6.6
P1	78, 79	6.8
P2	80, 81, 82, 100, 102	7.1
QUOTIENT OF	8, 9, 12, 14	2.1, 2.2, 2.3, 2.4
Q1	76, 79	6.8
Q2	1, 2	1.1
Q3	118, 121	8.2

Symbol	Expression No(s).	Simple Sentence No(s).
R	77, 78	6.8
READER	106	8.1
REASON FOR	108, 124	8.1, 8.2
RH	2, 21, 27, 29, 39, 58	1.1, 3.1, 4.1, 5.1, 5.2
RHO1	37	5.1
RI	5, 40, 77, 82, 108, 126	2.1, 5.1, 6.8, 7.1, 8.1, 8.2
RIVERS	73	6.7
RUGGED	69	6.5
RUSSIANS	86	7.1
S	73, 74	6.7
SEAS	53	5.2
SECOND	106, 112, 114, 118, 121	8.1, 8.2
SET OF ALL PEOPLE	80	7.1
SIZE OF	10, 11, 13, 15	2.2, 2.3, 2.4
SUBSET	3, 22, 30, 43, 51, 52, 53, 54, 59, 60, 79	1.1, 3.1, 4.1, 5.1, 5.2, 6.8
SUNLIT	20	3.1
SUNRISE	26	4.1
SUNSET	28	4.1
SUPERLATIVE	23, 125	4.1, 8.2
T	71, 72	6.6
TATARS	92	7.1
THIRD	107, 119	8.1, 8.2
TIA	44, 49	5.2
TIE	45, 49	5.2

Symbol	Expression No(s).	Simple Sentence No(s).
TIME ZONES	5	2.1
TIMES	22	3.1
TIR	19, 20, 24, 61, 63, 65, 67, 69, 71, 73, 75, 77, 119	3.1, 4.1, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 8.2
TIV	25, 26	4.1
TOUCHES	57	5.2
TUNDRA	61	6.1
UDMURTS	93	7.1
UKRANIANS	87	7.1
UNION	49, 50	5.2
USA	10, 31	2.2, 5.1
USSR	1, 4, 11, 16, 19, 57, 107, 112, 114	1.1, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 5.2, 8.1, 8.2
UZBEKS	99	7.1
VISITS	105	8.1
VLADIVOSTOK	25	4.1
VOLCANOES	65	6.3
V1	69, 70	6.5
WBR	24, 28	4.1
WEST OF	23	4.1
WESTERN-MOST	23, 24	4.1
WIDTH OF	4, 5, 16	2.1, 2.5
WIDTH OF USSR	4, 8	2.1

Symbol	Expression No(s).	Simple Sentence No(s).
W1	67, 68	6.4
W2	7, 8	2.1
W3	28, 29	4.1
W4	34, 35	5.1
W5	109, 110	8.1
X1	61, 62	6.1
X2	20, 21	3.1
X3	26, 27	4.1
X4	33, 34	5.1
X5	104, 105, 106, 107	8.1
Y1	63, 64	6.2
Y2	5, 6	2.1
Y3	21, 22	3.1
Y4	27, 30	4.1
Y5	32, 34	5.1
Y6	104, 108	8.1
Z1	65, 66	6.3
Z2	6, 7	2.1
Z3	29, 30	4.1
Z4	31, 32	5.1
Z5	108, 109, 126	8.1, 8.2

Symbol	Expression No(s).	Simple Sentence No(s).
2.5	9	2.2
7	12	2.3
12	38, 55, 56	5.1, 5.2
13	40	5.1
60	14	2.4
100	85	7.1
100000	74	6.7
225000000	81	7.1

Chapter 3
Deductive Procedures

3.1 A Question-Answering Procedure

We can conceive of a hierarchy of question-answering procedures, from very simple ones, which handle a limited set of true-false questions, to very elaborate ones that answer complicated "why" questions.

The procedure given here is very simple. It does, however, present a number of the concepts in our approach.

We assume that we are given a relational structure which serves as our "data-base", and a certain expression which is interpreted as a True-Falue question. Procedure 1 will produce the answer "yes" or "I don't know". Procedure 1 will not answer "no".

The procedure is based on two ideas. First, that we will work with a small subset of the data-base; this subset will be called the relevancy class. Second, that we will systematically enlarge this relevancy class until it contains the desired expression. The relevancy class is enlarged by the application of certain rules, namely the Enlargement Rule and any number of Deductive Rules.

The enlargement rule assigns expressions already in the data-base to the relevancy class. The deductive rules add new expressions (i.e., expressions not in the data-base) to the relevancy class.

Procedure 1 is now as follows.

1. Apply the enlargement rule to the statement of the question. This creates the first relevancy class.

2. If possible, apply a deductive rule to the relevancy class. This adds an expression to the relevancy class.

3. Repeat step 2 as often as possible. When no longer possible, apply the enlargement rule to the present relevancy class, creating thereby an enlarged relevancy class. Return to step 2.

There are two eventual outcomes: (a) The desired expression becomes a member of the relevancy class; (b) The desired expression does not become a member of the relevancy class and you become tired.

If (a) happens before (b), output the answer "yes". If (b) happens before (a), output the answer "I don't know".

3.2 The Enlargement Rule and Six Deductive Rules

Procedure 1 for answering T-F questions (see Section 3.1) refers to the enlargement rule. The rule is applied to a given relational structure expression or to a set of expressions.

The rule is as follows:

1. Form a list of all atoms occurring in the given expression or set of expressions.

2. Select from the data-base all expressions containing atoms on the list formed in step 1.

3. Append the set of expressions formed in step 2 to the relevancy class.

Procedure 1 for answering T-F questions (see Section 3.1) also refers to certain deductive rules. Six such rules are given below. The rules are understood to refer only to expressions in the relevancy class.

1. Rule of Subset.
Given the expression

$$(\subset, A, B)$$

for any A and B. List all explicit members* of A. List all explicit members of B. For any member, a_1, on the list of A which is not on the list of B, add the expression

---

* X is an explicit member of A if the relevancy class contains the expression (A,X) (or (ϵ,X,A)).

(B, a_i)

to the relevancy class.

2. First Rule of LI.

Given the expression

$(\underline{LI}, A,B,C)$

for any A, B, C. List all explicit members of A. List all explicit members of B. For any member, α_i, on the list of A which is not on the list of B, add the expression

(B, α_i)

to the relevancy class.

Note that α_i is an ordered pair.

3. Rule of RH.

Given the expression

$(\underline{RH}, A,B)$

for any A and B. List all explicit members of A. List all explicit members of B. If any member, a_i, of the list of A is not the right-hand member of a member of B, add the expression

(B, x_1, a_i)

to the relevancy class. Here x_1 is the first available (non-used) atom.

4. Second Rule of LI.
Given the expression

(<u>LI</u>, A,B,C)

for any A, B, C. List all explicit members of A (these are pairs). List all explicit members of C. Form the "left half" of the list of A; i.e., list all left-hand members of the pairs in the list of A. For any member a_i, in the list of the left half of A, which is not on the list of C, add the expression

(C, a_i)

to the relevancy class.

5. Rule of MINISET.
Given the expression

(<u>MINISET</u>, A, Ø, B)

for any A and B (Ø is the empty set). List all explicit members of A. Delete B from the list, wherever it occurs. For each remaining member, say a_i, add the expression

(=, a_i, B)

to the relevancy class.

6. Rule of Equals.
Given the expression

(=, A, B)

for any A and B, and any <u>other</u> expression E which contains A (or B). Append to the relevancy class the expression E' which is like E except for having each instance of A (or B) replaced by B (or A).

<u>3.3 An Example of Procedure 1</u>

The following illustrates the use of Procedure 1 for answering T-F questions (see Section 3.1), including the use of the enlargement rule and the first six deductive rules (see Section 3.2).

We assume we are given the following data-base (which translates into English as "Russia is the biggest country. Japan is a country.")

(<u>LI</u>, X, BIGGER, Y)
(<u>MINISET</u>, Y, Ø, RUSSIA)
(<u>RH</u>, Z, X)
($\subset$, COUNTRIES, Z)
(COUNTRIES, JAPAN)

We assume further that we are presented with the following question (which translates into English as "Is Russia bigger than Japan?")

(BIGGER, RUSSIA, JAPAN)?

<u>Step 1</u>. We apply the enlargement rule to the <u>question</u>, and obtain the following relevancy class:

(<u>LI</u>, X, BIGGER, Y)
(<u>MINISET</u>, Y, Ø, RUSSIA)
(COUNTRIES, JAPAN)

<u>Step 2</u>. We try to apply the deductive rules, but all of them fail.

Step 3. We apply the enlargement rule to the present relevancy class. This adds two more expressions. Our relevancy class is now:

(LI, X, BIGGER, Y)
(MINISET, Y, $\emptyset$, RUSSIA)
(COUNTRIES, JAPAN)
(RH, Z, X)
($\subset$, COUNTRIES, Z)

Step 4. We apply Rule of Subset to

($\subset$, COUNTRIES, Z)

List members of COUNTRIES: JAPAN
List members of Z: none.
Hence, append expression

(Z, JAPAN)

to the relevancy class. Our relevancy class is now:

(LI, X, BIGGER, Y)
(MINISET, Y, $\emptyset$, RUSSIA)
(COUNTRIES, JAPAN)
(RH, Z, X)
($\subset$, COUNTRIES, Z)
(Z, JAPAN)

Step 5. We apply Rule of RH to

(RH, Z, X)

List all members of Z: JAPAN

List all members of X: none

Hence, add expression

$$(X,\ x_1,\ \text{JAPAN})$$

to the relevancy class.

The relevancy class is now:

(<u>LI</u>, X, BIGGER, Y)
(<u>MINISET</u>, Y, Ø, RUSSIA)
(COUNTRIES, JAPAN)
(<u>RH</u>, Z, X)
($\subset$, COUNTRIES, Z)
(Z, JAPAN)
(X, x_1, JAPAN)

<u>Step 6</u>. We apply the First Rule of <u>LI</u> to

(<u>LI</u>, X, BIGGER, Y).

List all explicit members of X: (x_1, JAPAN)

List all members of BIGGER: none

Hence, add expression

(BIGGER, x_1, JAPAN)

to the relevancy class.

The relevancy class is now:

(<u>LI</u>, X, BIGGER, Y)
(<u>MINISET</u>, Y, Ø, RUSSIA)
(COUNTRIES, JAPAN)
(<u>RH</u>, Z, X)
($\subset$, COUNTRIES, Z)
(Z, JAPAN)
(X, x_1, JAPAN)
(BIGGER, x_1, JAPAN)

<u>Step 7</u>. We apply the Second Rule of <u>LI</u> to

(<u>LI</u>, X, BIGGER, Y)

List all members of X: (x_1, JAPAN)
Form their left half: x_1
List all members of Y: none.
Hence, add expression

(Y, x_1)

to the relevancy class.

The relevancy class is now:

(<u>LI</u>, X, BIGGER, Y)
(<u>MINISET</u>, Y, Ø, RUSSIA)
(COUNTRIES, JAPAN)
(<u>RH</u>, Z, X)
($\subset$, COUNTRIES, Z)
(Z, JAPAN)
(X, x_1, JAPAN)
(BIGGER, x_1, JAPAN)
(Y, x_1)

Step 8. We apply the Rule of MINISET to

(MINISET, Y, Ø, RUSSIA)

List all members of Y: x_1
Delete RUSSIA from list.
We are left with x_1.
Hence, add the expression

(=, x_1, RUSSIA)

to the relevancy class.

The relevancy class is now:

(LI, X, BIGGER, Y)
(MINISET, Y, Ø, RUSSIA)
(COUNTRIES, JAPAN)
(RH, Z, X)
($\subset$, COUNTRIES, Z)
(Z, JAPAN)
(X, x_1, JAPAN)
(BIGGER, x_1, JAPAN)
(Y, x_1)
(=, x_1, RUSSIA)

Step 9. We apply the Rule of Equals to

(=, x_1, RUSSIA)

We can apply the rule four times, adding the following four expressions to the relevancy class.

($\underline{\text{MINISET}}$, Y, $\emptyset$, x_1)
(X, RUSSIA, JAPAN)
(BIGGER, RUSSIA, JAPAN)
(Y, RUSSIA)

At this point we are <u>done</u>, since we have the desired expression

(BIGGER, RUSSIA, JAPAN)

in our relevancy class.

The answer to the question is "yes".

References

1. Green, Bert, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery: "BASEBALL: An Automatic Question-Answerer", Proceedings of the 1962 Western Joint Computer Conference, pp. 219-224. Reprinted in Computers and Thought edited by Edward Feigenbaum and Julian Feldman, McGraw-Hill Book Company, New York, 1963.

2. Lindsay, Robert K: "Inferential Memory as the Basis of Machines Which Understand Natural Language", Computers and Thought edited by Edward Feigenbaum and Julian Feldman, McGraw-Hill Book Company, New York, 1963, pp. 217-233.

3. Simmons, Robert S.: "Answering English Questions by Computer: A Survey", Communications of the ACM, Vol. VIII, 1, January 1965, pp. 53-70.

4. Bertram Raphael: SIR: A Computer Program for Semantic Information Retrieval, June 1964, TR 2 Project MAC, MIT.

5. R. Levien and M.E. Maron: Relational Data File, RM-4793-PR, The RAND Corporation, December 1965.

6. J.A. Feldman: Aspects of Associative Processing, Technical Note 1965-13, MIT Lincoln Laboratory, Lexington, Massachusetts.

7. A.D. Savitt, H.H Love, R.E. Troop: Association-Storing Processor Study, RADC-TR-66-174, June 1966, Rome Air Development Center.

8. C. Green and Bertram Raphael: "Research on Intelligence Question-Answering System", Scientific Report 1, Stanford Research Institute, Menlo Park, California, May 1967.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Computer Corporation of America 565 Technology Square Cambridge, Mass. 02139	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A	
3. REPORT TITLE FUNDAMENTALS OF RELATIONAL STRUCTURES		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) None		
5. AUTHOR(S) (First name, middle initial, last name) None		
6. REPORT DATE 3 September 1968	7a. TOTAL NO. OF PAGES 62	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. AF 19(628)-5939 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-68-404 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Command Systems Division, Electronic Systems Division, Air Force Systems Command, USAF, L G Hanscom Field, Bedford, Mass. 01730	

13. ABSTRACT

A system of notation, having very elementary syntax, is developed. It is shown by example how arbitrary information may be represented in this notation. The notation is intended for computer processing, and certain deductive procedures, suitable for computers, are outlined.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer						
Natural language						
Deductive procedure						